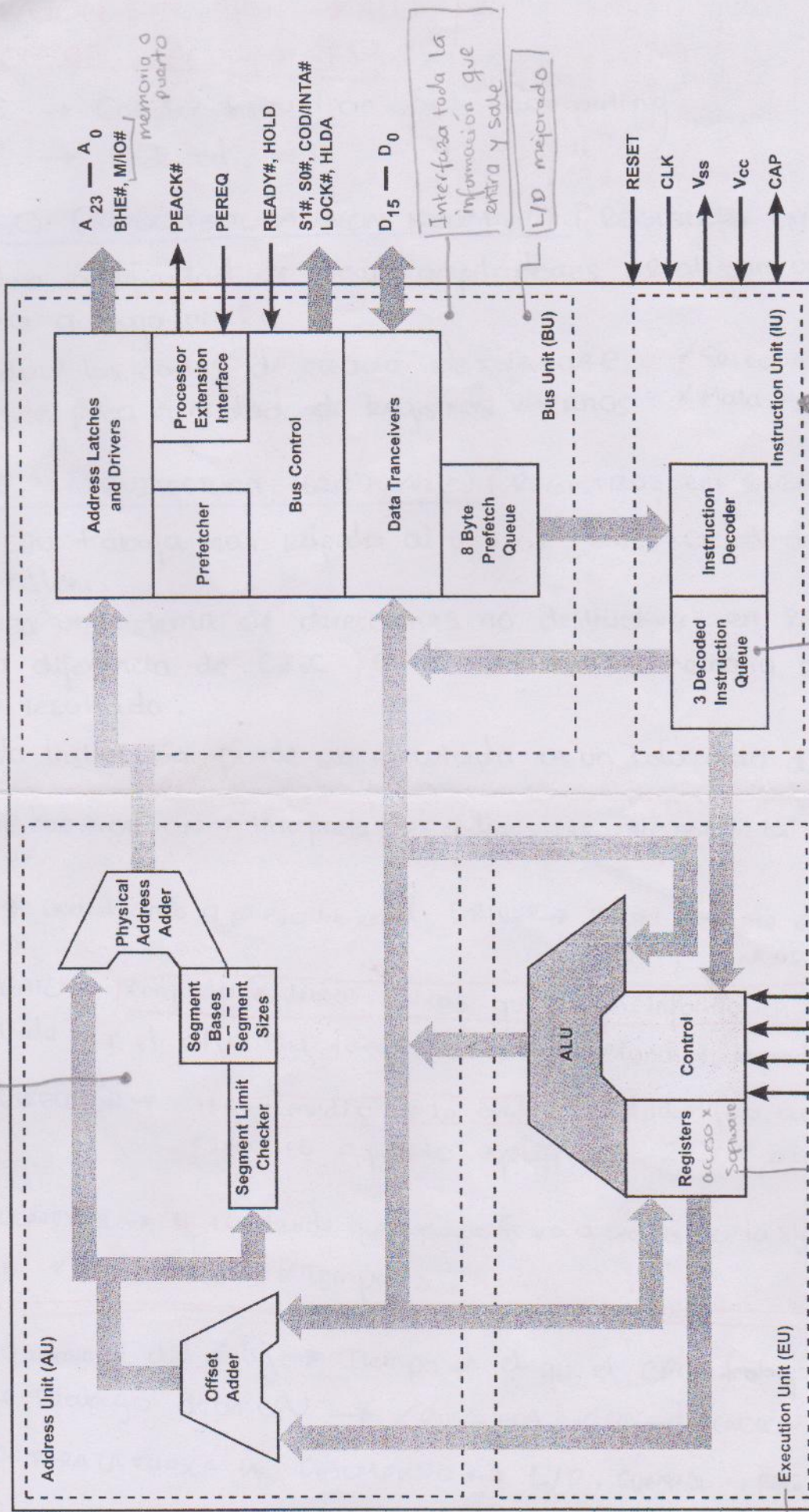


Intel 80286 architecture



Servicio para interpretar la idea y valores en tablero binario para poder llevarla a cabo

Unidad de Codificación

Interpreta los códigos de operación

es la antigua a la cache, FIFO, almacena instr. digital q le dice a la unidad de control q hacer

tambien temporales

No es donde se guarda todo, se llama así por el ALU.

UNIDAD DE OPERACION DEL CPU → ALU
UNIDAD DE DIRECCION DEL CPU → IP

CISC → Complex Instruction Set Computing
RISC → Reduced " " "

CISC (ARQUITECTURA VON NEUMAN) [REGISTROS EN SERIE]

- ✓ Reduce la dificultad de crear compiladores, realizan un salto semántico corto a bajo nivel.
- ✓ Reduce los costos de creación de software
- ✓ Posee poca cantidad de registros internos
- ✓ Se calienta más que RISC
- ✓ Mala calidad de buses

RISC (ARQUITECTURA HARDWARE) [REGISTROS EN PARALELO]

- ✓ LA CPU trabaja más rápido al utilizar menos ciclos de reloj para ejecutar instrucciones
- ✓ Utiliza un sistema de direcciones no destructivas en RAM. Esto significa que a diferencia de CISC, RISC conserva en memoria sus dos operandos y el resultado.
- ✓ Cada instrucción puede ser ejecutada en un solo ciclo del CPU.

UNIDAD DE CODIFICACION → Interpreta los códigos que representan la orden que se va a ejecutar.

LATCH → Guarda info q se escribe en el DRIVER → Drena corriente dentro en forma de amperaje, que no dañen lo de adentro

BUS DE DATO → Conjunto de líneas físicas que llevan información digital que será computada por el CPU. Del tamaño de este bus (ancho) depende la calidad del CPU

BUS DE DIRECCION → Por medio de la cual el CPU identifica una celda de memoria en específico o puerto específico $2^k = \#$ o cont max de mem en celdas

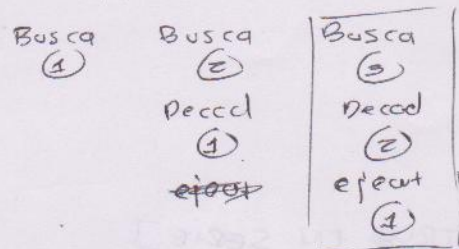
BUS DE CONTROL → Desde que operación va a realizar la sincroniza y temporiza
✓ Definir ✓ Sincronizar ✓ Temporizar

frecuencia nominal del reloj → Tiempo en el que el CPU trabaja Optimamente

Ciclo de Trabajo de un CPU → ✓ BUSQUEDA ✓ DECODIFICACION ✓ EJECUCION (Sub ciclos)

BLOQUES PARA LA TABLA DE DESEMPEÑO → L/D, CONTROL, REG. TEM, REG. A SOFTWARE
DECODIFICACION, ALU

Se dice que intel realiza los 3 sub-ciclos en un sólo periodo de reloj



Realizando los 3 sub-ciclos en un solo periodo de reloj para 3 diferentes instrucciones

Existen 4 lotes de Registros

(1) lotes de registros de propósito General // aparecen por todos lados en la sintaxis a bajo nivel por eso se les llama así
 Registro Acumulador, Registro base, Registro Contador (Arg CISC)
 Registro de Datos

(2) lotes de registros de Índices y apuntadores

(3) " " " " Segmento

(4) " " " " Mantenimiento

word = (16 bits)

doble word = (32 bits)

byte = (8 bits)

parrafo = (16 celdas) 16 bytes
~~16 bytes~~

(1)	AH	AX	AL	Acumulador
	BH	BX	BL	Base
	CH	CX	CL	contador
	DH	DX	DL	dato
	15 ← → 0			

(2) Indices memorias aleatorias = destino
 apuntad. " estáticas = fuente
 " " " " base
 " " " " LIFO

SI Reg índice fuente
 DI " " Destino
 BP Apuntador Base
 SP " Stack

← 1 word →

(3) → modo protegido (Reg de Sectores)
 → " Real (" " Segmento)

CS	Reg de Seg de código
DS	" " " " Datos
ES	" " " " extra
SS	" " " " stack

← 1 word →

(4) IP Reg Apuntador de Instrucción
 IF " status o bandera

← 1 word →

REGISTROS BANDERA

D = Director T = Trap
 I = Interruptore

OV = overflow

AC = carry auxiliar

S = sign

P = parity

C = carry

Z = zero

MODO DE DIRECCIONAMIENTO DE DATO

→ REGISTRO MOV AX, BX
 → INMEDIATO MOV BL, 3A
 → DIRECTO MOV[1234], AX
 → INDIRECTO POR REGISTRO MOV[BX], AX

→ A LA MEMORIA DEL PROGRAMA DIRECTO JMP CSN, IPN

→ A LA MEMORIA DEL PROGRAMA RELATIVO
 JMP e offset
 JMP Flag, e

→ BASE MAS INDICE MOV[BX+SI], AX

→ RELATIVO POR REG MOV[BX+4], AX

XOR[SI, FEED]
 NOT[DI, BE]

→ RELATIVO A LA BASE MAS INDICE

MOV ARRAY[BX+SI], AX
 ARRAY EQU 0x50
 MOV[BX+SI, array], AX
 offset

→ A LA MEMORIA DEL PROGRAMA INDIRECTO

JMP Registro
 JMP [Registro]

→ A LA MEMORIA DE LA PILA

PUSH Registro
 POP Registro
 CALL etiqueta
 RETURN

Coca :3

